ALTERING UAV FLIGHT PATH BY THREATENING COLLISION

Pietro Pierpaoli, University of Miami, Coral Gables, FL
Magnus Egerstedt, Georgia Institute of Technology, Atlanta, GA
Amir Rahmani, Jet Propulsion Laboratory, Pasadena, CA

Abstract

The ongoing transformation of air traffic control towards decentralized decision making based on ADS-B information shared by neighboring traffic will allow all aircraft and UAS in particular, to automatically detect and resolve collisions. In this work we highlight the importance of trustworthiness in such distributed systems, showing that autonomous aircraft can be forced into predetermined trajectories when their precise position and velocity are available to a potentially malicious craft. In other words, *malicious pursuer* players (real or hoaxed) taking advantage of shared data and collision avoidance properties, can dictate *evader* agent trajectory, which might not realize the threat at all. As shown by numerical simulations and ground robot experiments, combination of arcs and straight paths can be achieved and be used to arbitrarily control the evader.

Introduction

The ongoing process of modernization in air traffic control (ATC) will transform current radar–based centralized architecture into a network based decentralized system useful for both military and commercial flights. A GPS data sharing protocol between nodes will substitute the outdated standard radar infrastructure providing precise aircraft localization in both air and ground operations. The well-known Automatic Dependent Surveillance-Broadcast (ADS-B) system will collect and share satellite based information on position and velocity with neighboring traffic and ground bases. As a result, an aircraft (UAS in particular) will be capable of automatically detect and resolve collisions. Next Generation (NextGen) Air Transportation Systems and Single European Sky ATM Research (SESAR) are intended to realize this change. Main goal is to overcome current limitations due to old regulations, scarce aircraft localization and mainly voice-based communications. Despite recent concerns and delays [2] [3] [4], ADS-B technology is well on its way to realization and could introduce many benefits in ATC. Expected benefits will affect route efficiency, airports improved inflow/outflow capacity and systematic tracking of weather status. Moreover, it can prove instrumental in integration of unmanned air systems with traditional manned vehicles. A thorough investigation of the system is the necessary pre-requisite to wide adaptation of these systems. Possible vulnerabilities have been recently studied. Accidental and involuntary malfunctioning of the system such as communication interruptions are only some of possible sources of hazard. Intentional actions conducted against system safety are still a big challenge even in their original definition. Spoofing, jamming or injection of fake agents are known example of GPS information sharing weakness [5], [1].

In this work, using pursuers-evader scenarios, we further investigate a different class of safety issues, strictly related to neighbor's trustworthiness, previously introduced by the authors [6] and extended in this paper. As already mentioned, collision avoidance techniques must be employed onboard autonomous vehicles in order to guarantee safety during all operations and also be used to detect hazardous situations on traditionally manned airplanes. Regardless of the particular method adopted, the presence of sense and avoid system in the control loop introduces a dependence on the external environment.

Enhanced by the aforementioned sharing of flying information framework, opposing (pursuers) UAVs can exploit this dependence and eventually alter a UAV's path as they wish. For the sake of simplicity and given arbitrary changes in altitude is unlikely due to regulatory and performance constraints, a 2-D scenario is considered. Once the aircraft dynamics are introduced, the presence of a sense-and-avoid layer can be represented by a switching controller triggered by a collision threat. In this framework, the evader represents a dynamical system whose trajectory is the pursuers' control objective. Controllability of this system is investigated and with the proposed strategy, two pursuers can obtain desired output in two sequential steps. In the first step pursuers perform an approaching maneuver in order to reach a specific distance from the evader. A model predictive controller along with a trajectory-tracking model has been implemented for this purpose. In the second step, by means of an open loop constant radius turn, pursuers force the evader into a turn itself. Once the desired heading is reached the pursuers can keep the evader in a straight flight or induce a second maneuver adopting the same procedure. Nonlinearity due to aircraft kinematic, switching control and collision avoidance method strongly preclude the possibility of closed-form solution for the problem. Nevertheless, as we will show, simple results can be obtained and reveal an interesting class of weaknesses in autonomous vehicles.

The rest of this paper is arranged as follows: in the next section environment and aircraft model is introduced in order to define our evader-pursuer scenario. Then, evader and collision avoidance algorithm are presented and synthetized in a single system. In the fourth section, we define pursuers strategy and conditions for evader controllability. Finally, we present the implementation of the proposed model in numerical and experimental simulations.

Aircraft Model and Game Scenario

In this section we define the working environment and aircraft model. As mentioned before, a sharing of satellite based information is the key feature in future air traffic system. Aircraft will communicate flight data information, such as position and velocity to neighbors and ground stations; therefore, we assume this set of information reliable and accessible by all neighbors at any time. Given current airspace altitude layered structure, and following from the low fuel efficiency in changes of flying velocity, we insist that collision must be resolved within flying plane with no changes in velocity. Therefore, denoting the state of the aircraft as $\boldsymbol{x} = \{x, y, \theta\}^T$, agent kinematic model can be represented using a constant speed unicycle model:

$$\dot{\boldsymbol{x}} = \begin{cases} \dot{x} = V\cos(\theta) \\ \dot{y} = V\sin(\theta) \\ \dot{\theta} = \omega \end{cases}$$

where x, y *and* θ represent position and heading on the plane, while V and ω are linear and angular velocity respectively. The description of the aircraft dynamics through a unicycle model is justified here by the in-plane flight assumptions and limited turning radius.

The set of information shared among agent can be summarized in a vector q:

$$q = \{x, y, \theta, V, \omega\}^T$$

which gives complete information on each neighbor's state and trajectory. Introducing a classic nomenclature used in game theory, we present our evader-pursuers scenario. The evader is defined to be an agent provided with a collision avoidance system, employing a flying information sharing protocol, whose target is to safely reach its final destination. On the other side, pursuer agents, relying on evader collision avoidance property, tries to impose a desired trajectory on the opponent. In order to do so, pursuer agents freely decide their control inputs and therefore they will not follow any collision avoidance reasoning.

Evader Model

In this section we introduce the evader model. Following from the role defined in the previous section, we introduce the collision avoidance technique and how it affects the agent's dynamic model.

Velocity Obstacle method

Collision detection and resolution in autonomous systems has been a canonical problem with many application fields resulting in a rich literature pool. The reader is referred to [7] and [8] for complete and detailed surveys on the topic. Without loss of generality Velocity Obstacle (VO) method [9] and some of its extensions [10] have been selected as base for the collisions resolution algorithm used in this work. VO's popularity, simple implementation and agreement with ADS-B sharing framework led to this choice. VO represents a geometric, decentralized and global collision avoidance method. Even though single collisions are treated separately, situations with more than one neighbor can be easily handled by overlapping results from single conflicts. Each aircraft (or agent) is modeled as a massless point centered at a circular protected area of radius r_{PZ}. The presence of other agents or obstacles inside the protected area is considered undesirable and will therefore treated as a collision. In what follows, we present informal and intuitive description of the method, limiting the presentation to the fundamental results required. Detailed proof can be found in [9]. With reference to Figure 1, let us consider two agents E and P having velocity V_E and V_P respectively and denote $\Delta V = V_E - V_P$ their vector difference. Let us also denote λ_1 and λ_2 the angles formed by the two tangents from E to agent P protected area with respect to a global coordinate system. Considering the triangle ΔV, V_E and V_P, a variation in any of these quantities will also result in variations in the others. In particular, when varying V_P such that ΔV spans the range $[\lambda_1, \lambda_2]$, vector V_E will span interval $[\beta_1, \beta_2]$.

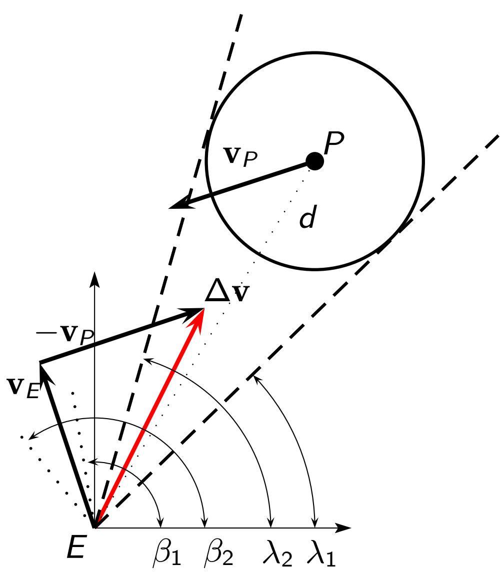

Figure **1**. Velocity Obstacle Scheme.

Accordingly to the original formulation of the method, as long as vector ΔV does not belong to the region delimited by the two tangents to the neighbor's protected area, the distance between the agents will not decrease below the value r_{PZ}, and therefore, no collision is expected to occur. Give the mapping between set $[\lambda_1, \lambda_2]$ and $[\beta_1, \beta_2]$, in order to avoid the collision, V_E must lay outside the region delimited by the lines with orientation β_1 and β_2. Therefore, the oriented set of angles $[\beta_1, \beta_2]$ marks an obstacle for the velocity vector. Note, in case of constant speed, the velocity obstacle corresponds to a constraint on vehicle heading. Therefore, with a slight abuse of notation, we define $\angle V$ as the angle formed by vector V. Therefore:

$$\angle V_E \notin [\beta_1, \beta_2] \Leftrightarrow \angle \Delta V \notin [\lambda_1, \lambda_2]$$

This relation guarantees a condition for a free from collision navigation and represents the main result on the Velocity Obstacle method.

Deconfliction policy

Following results from previous section, if aircraft heading belongs to the set of obstacle (collision) angles $[\beta_1, \beta_2]$, at some point in the future the distance between agents will drop below the safety distance r_{ZP}. Once the criteria for identify undesired headings has been defined, we need to introduce the set of decision rules that will maintain a safe navigation. For the sake of generality, in what follows, we assume non-cooperative agents. Therefore, each of the aircraft involved in the conflict will perform an evading maneuver without assuming changes in the opponent's path. With reference to Figure 2, let us first assume aircraft heading (θ) does not belong to the set of angles $[\beta_1, \beta_2]$. In that case, it is possible to maintain the original desired direction (θ_d) towards the goal.

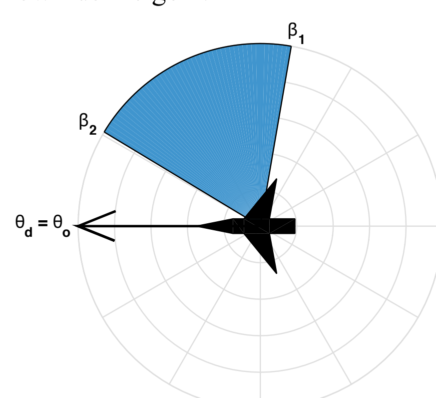

Figure 2. Available Original Headings.

Assume now, that after a certain amount of time, the set of forbidden heading $[\beta_1, \beta_2]$ overlap the original flying direction, as presented in Figure 3. In this case, in order to avoid loss of minimum separation, a turning maneuver must be executed until a safe heading is reached.

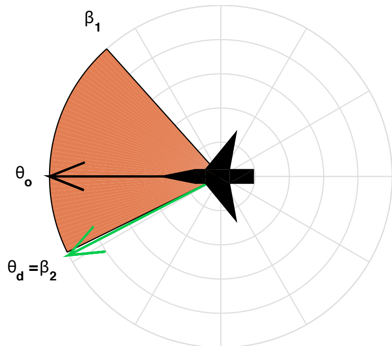

Figure 3. Non-available original heading. Direction β_2 Chosen Instead.

In general, we assume a least perturbation approach, that is, whenever the desired heading belongs to the set [β_1, β_2], the aircraft will try to resolve the conflict minimizing the perturbation from its original plan. Hence denoting by θ_o the original heading and by θ_d the desired one:

$$if\ \theta_o \in [\beta_1, \beta_2]\quad then\quad \theta_d = \hat{\beta}$$

where:

$$\hat{\beta} = \begin{cases} \beta_1 & if\ \|\theta_o - \beta_1\| < \frac{\|\beta_2 - \beta_1\|}{2} \\ \beta_2 & if\ \|\theta_o - \beta_2\| < \frac{\|\beta_2 - \beta_1\|}{2} \end{cases}.$$

Switching control model

Given the deconfliction policy presented in the previous section, the dynamic model for the autonomous agent can be modified as follows. Since no changes in linear velocity are considered, accordingly to the presented model, the control of evader movement is represented by the angular velocity ω. Assuming a proportional controller for ω with constant K_P, the evader control model is:

$$\dot{\boldsymbol{x}}_{\boldsymbol{E}} = \begin{cases} \dot{x}_E = V_E \cos(\theta_E) \\ \dot{y}_E = V_E \sin(\theta_E) \\ \dot{\theta}_E = \omega_E = \begin{cases} K_P(\theta_d - \hat{\beta}) & if\ \theta_o \in [\beta_1, \beta_2] \\ K_P(\theta_d - \theta_o) & if\ \theta_o \notin [\beta_1, \beta_2] \end{cases} \end{cases}$$

Noting the dependence between velocity obstacle [β_1, β_2] and neighbors' velocity, the dependence of evaders' trajectory to position and velocity of neighbors is evident.

Pursuers model and Strategy

In this section we introduce pursuers model and the strategy adopted to exploit evader's sense and avoid algorithm. Given the aircraft's first order model, the pursuer dynamics can be expressed as:

$$\dot{\boldsymbol{x}}_{\boldsymbol{P}} = \begin{cases} \dot{x}_P = V_P \cos(\theta_P) \\ \dot{y}_P = V_P \sin(\theta_P) \\ \dot{\theta}_P = \omega_P \end{cases}$$

As mentioned in the previous section, pursuer agents do not employ any collision avoidance method. Their purpose is to bring evader from its initial state to a desired one, performing a specific trajectory. The values of pursuer controls, that is, linear and angular velocity (V_P and ω_P), define the particular forcing trajectory.

Pursuer agents' strategy relies on the fact that their position and velocity create specific obstacles velocity for the evader. Since each pursuer creates a single velocity obstacle, their number also has consequences on the result. As shown in Figure 4, two pursuers can be used to arbitrarily limit evader velocity. In such a scenario, regardless of the particular deconfliction policy applied, evader autonomy can be arbitrarily reduced.

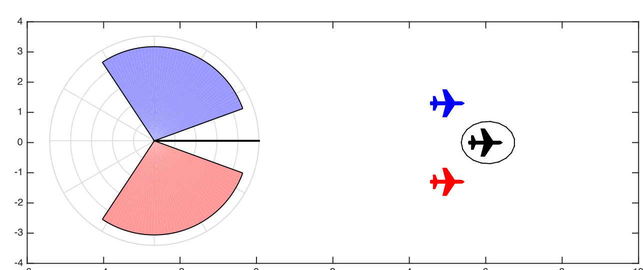

Figure 4. Evader Velocity Obstacle in Case of Two Neighboring Pursuers.

Forcing strategy definition

Given kinematic model, collision avoidance algorithm and evader switching control the resulting complete control system is highly non-linear. For this reason, deriving a closed form solution for the evader-pursuer trajectory problem and eventually proving its controllability is not a trivial problem. Nevertheless, assuming the forcing maneuver as a composition of constant radius turns and straight paths, important and motivating results can be obtained.

Forcing Phase

The most intuitive and simple trajectory is a constant speed (V_F), fixed rate (ω_F) turning maneuver (Figure 5). Numerical simulations show this motion primitive can successfully be used to force evader's heading towards a different direction. In particular, a combination of circular and straight paths can be used to induce variations in the evader route, whose response is accorded by the collision avoidance algorithm. Particular values of V_F and ω_F must be chosen to comply with evader performance, and once defined, the complete turning maneuver is executed in an open loop fashion.

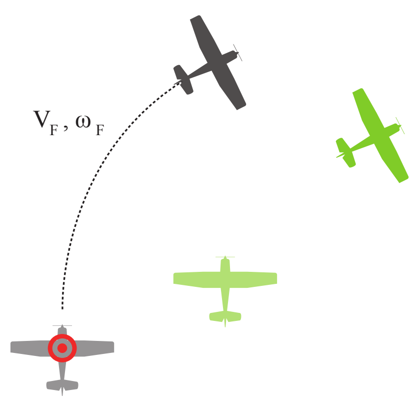

Figure 5. Forcing Maneuver. Circular Maneuver Conducted at Defined V_F and ω_F.

Approaching Phase

In order to succeed in its intent, pursuer agents must reach a certain relative position from the evader before it can start changing its flight path. We define this configuration as the pair $\Gamma: \{\delta, \rho\}$ composed by the position in a polar coordinate system centered in evader position and oriented along its velocity vector (Figure 6). Therefore, δ is radial distance from the evader and ρ is the angular coordinate.

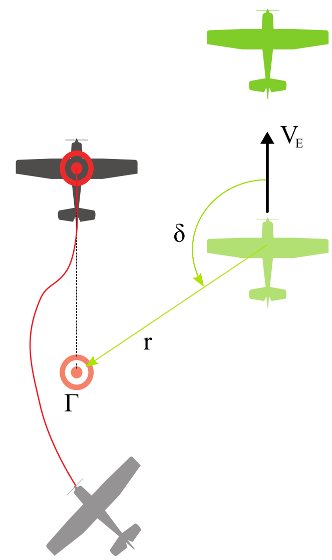

Figure 6. Approaching Maneuver. Pursuer (black) Reaches Desired Position in Evader (green) Reference Frame.

Assuming assigned trajectory parameters (V_F, ω_F), the correct initial configuration Γ is defined using a recursive numerical simulation.

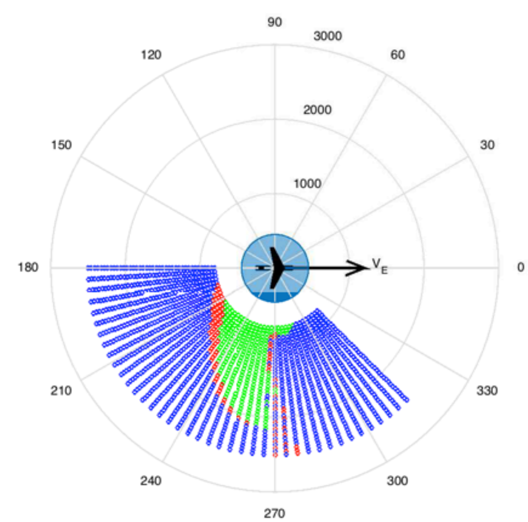

Figure 7. Outcomes for Different Starting Configurations. Ineffective (blue), colliding (red), effective configurations (green).

In the example presented in Figure 7, the evader in the center in flying straight towards its goal located at 0π and pursuers goal is to impose a turning maneuver towards $\pi/2$.

As showed in Figure 7, different values of Γ are marked accordingly with their effect on evader trajectory. Forcing maneuvers started in blue positions have none or partial effect on evader trajectory, those started in red cause unavoidable collision. Positions marked in green represent correct starting configurations to steer the evader towards $\pi/2$ given assigned values of V_F and ω_F.

Trajectory Tracking Formulation

Considering results from the previous section, pursuer agents must reach a specific configuration in order to gain the required authority on evader's control. In particular, this configuration, defined by Γ, is a target position moving along a trajectory equal to evader's trajectory. For this reason, dynamics of the target configuration can be obtained knowing evader control inputs. We define now a trajectory-tracking problem, whose solution provides controls required to drive pursuers to the desired configuration. The state of the reference configuration is the vector $\boldsymbol{x_\Gamma} = [x_\Gamma, y_\Gamma, \theta_\Gamma]$; it is possible to describe its movements using the same first order differential model used previously:

$$\dot{\boldsymbol{x}}_{\boldsymbol{\Gamma}} = \begin{cases} \dot{x}_\Gamma = V_\Gamma \cos(\theta_\Gamma) \\ \dot{y}_\Gamma = V_\Gamma \sin(\theta_\Gamma) \\ \dot{\theta}_\Gamma = \omega_\Gamma \end{cases}$$

where V_Γ and ω_Γ are the same evader input controls obtained from the ADS-B sharing protocol. Given the definition of the reference state $\boldsymbol{x_\Gamma}$ and following a similar procedure presented in [11] and repeated in [6], it is possible to define the state error $\boldsymbol{x}_E$ as:

$$\boldsymbol{x_E} = \boldsymbol{T_\Gamma} \begin{pmatrix} x_P - x_\Gamma \\ y_P - y_\Gamma \\ \theta_P - \theta_\Gamma \end{pmatrix}$$

where $\boldsymbol{T_\Gamma}$ is the rotation matrix required to express $\boldsymbol{x_\Gamma}$ in the reference system of $\boldsymbol{x_P}$ (see Figure 8).

From the definition of the error state, we obtain a first order differential system for the relative dynamics, in the form:

$$\dot{\boldsymbol{x}}_E = f(\boldsymbol{x}_E, V_P, \omega_P, V_\Gamma, \omega_\Gamma)$$

Error trajectory is obviously defined as function of evader and pursuers control inputs. A model predictive control (MPC) has been formulated and the control inputs are obtained from the respective minimization problem. Note that in order to obtain a solution (approaching phase control inputs), pursuers must have better performance than evader; i.e. be more dynamically agile.

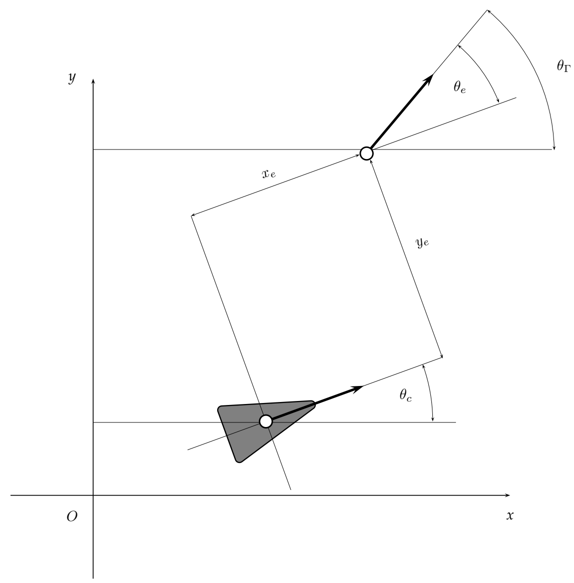

Figure 8. Configuration Error State and Reference System Transformation.

Performance constraints and error dynamic model represent constraints in the optimization problem, while the minimization objective function introduces costs associated to the control efforts (V_P, ω_P) and the error ($\boldsymbol{x}_E$) over the finite horizon τ. Therefore, complete formulation of optimization problem is the following.

Minimize:

$$J = \sum_{i=1}^{\tau} \boldsymbol{x}_{E,i}{}^T Q \boldsymbol{x}_{E,i} + V_{P,i}{}^T R V_{P,i} + \omega_{P,i}{}^T S \omega_{P,i}$$

subject to:

$$\begin{aligned} \dot{\boldsymbol{x}}_{E,i} &= f(\boldsymbol{x}_{E,i}, V_{P,i}, \omega_{P,i}, V_{\Gamma,i}, \omega_{\Gamma,i}) \\ V_{P,i} &\in \mathcal{V} \qquad\qquad i = 1, \dots, \tau \\ \omega_{P,i} &\in \mathcal{W} \end{aligned}$$

Matrix Q, R and S are opportune weight matrices.

Implementation

In the previous section a strategy to induce a change in a UAV path by threatening a collision has been presented. Exploit of sense and avoid properties eventually allows to control an autonomous vehicle directly affecting its navigation parameters. As previously mentioned, given the high non-linearity of the complete control problem, rigorous proofs are hard to be defined. Nevertheless, as we show in this section, both numerical and experimental results reveal the validity of our approach.

In both simulations and experiments, the approach-force-hold strategy presented earlier is implemented. Pursuers goals are defined as particular flying directions $\tilde{\theta}$ for the evader. The forcing maneuver is considered successful when $\theta_E = \tilde{\theta}$. The formulation and solution of the MPC introduced in the previous section has been formulated in YALMIP [12] and solved using MATLAB built-in solver. Both simulations and experiments are conducted using two pursuers. In this case, it is possible to create an arbitrarily narrow set of available headings for the evader and strongly limits its autonomy.

Numerical Simulations

We first present numerical simulation of a series of sequential complete forcing maneuvers. The complete simulation is composed by three separate forcing maneuvers. After the approaching maneuver (Figure **9**), pursuers (in red) aim to steer evader (black) until a heading equal to 7/6π is reached (Figure 10). At this point, both pursuers fly a straight path, maintaining evader's velocity (Figure 11). The second forcing maneuver brings the evader to a heading equal to 3/4π (Figure 11).

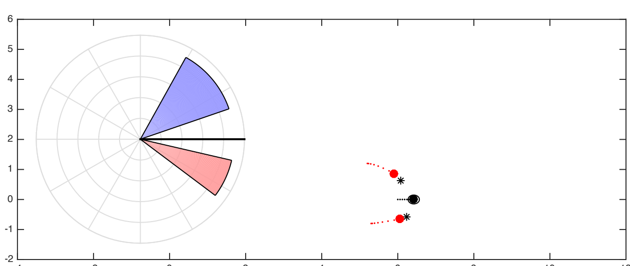

Figure 9**. Approaching Maneuver.**

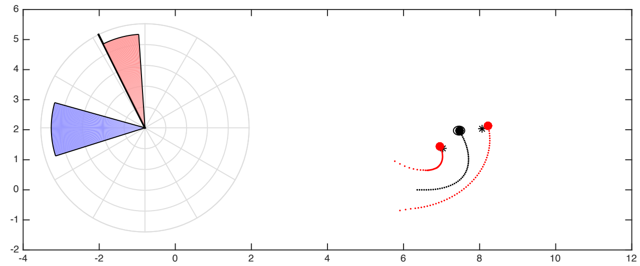

Figure 10. First Forcing Maneuver To 7/6π.

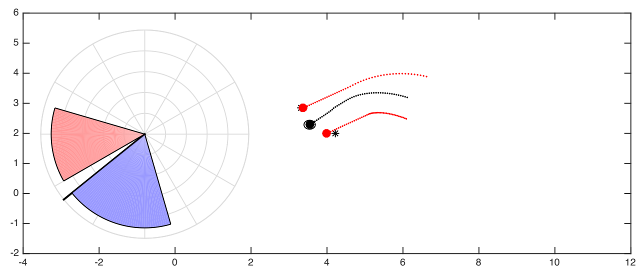

Figure 11. Holding Maneuver.

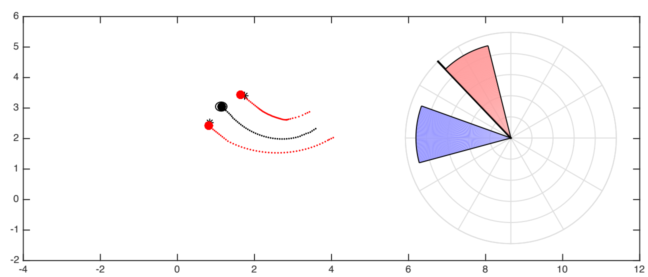

Figure 12. Second Forcing Maneuver to 3/4π.

The last maneuver performed keeps the evader inside a prescribed circle with no chances of escape (Figure 12). During the whole simulation, evader final goal is the initial one, located at *(L, 0)*, where *L* is a large number.

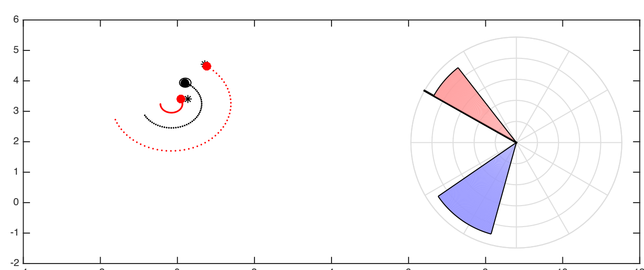

Figure 13. Third Forcing Maneuver. Evader is Locked Inside a Prescribed Circle.

Robot Simulation

Real robot simulations have been conducted using three Khepera III from K-team. Khepera III is a differential drive wheels robot with 600MHz ARM processor, 128Mb RAM and wireless card for remote communication. An indoor tracking camera system is used to determine agents position. Ten OptiTrack S250e motion capture cameras were used. Processing of tracking information, collision avoidance and control parameter definition were performed using MATLAB on a Windows 7 PC. Control signal to the robots were then communicated to the robots via wireless communications.

In accordance with the premise of this work, only information on positions is shared among agents. From Figure 14 to Figure 16 we present results from a one-evader/two-pursuers scenario. Results presented are snapshots taken from a camera mounted on the ceiling pointing downwards. Graphic elements on the floor are real-time simulation data obtained using a projector mounted on the ceiling and pointing downwards. With reference to Figure 13 to Figure 15, the reference systems used in the experiments has *x*-axis pointing down and y-axis pointing right. The central robot plays evader role and the black circle around it represents its protected zone. Red and blue sectors represent the obstacle velocities associated with red and blue pursuers respectively. Red mark in the center represents the center of the reference system while the four red marks in the corner delimit the tracking system working area. These marks are not projected features.

Figure 14. Approaching Maneuver in Robot Simulation.

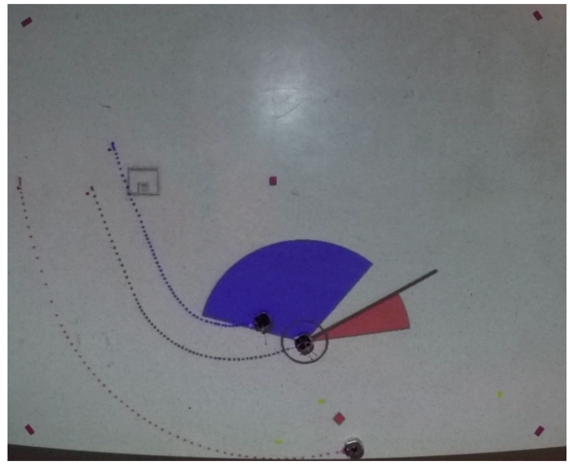

Figure 15. Forcing Maneuver in Robot Simulation.

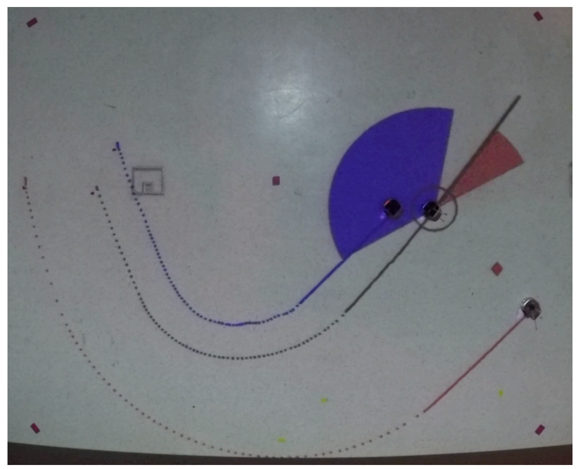

Figure 16. Holding Maneuver in Robot Simulation.

Evader goal is located straight respect to its initial heading, at a virtual distance of 300 meters. Pursuers aims to impose a new $3/4\pi$ heading on the evader. Similar to previous simulations, the pursuers first reach initial target positions Γ (red diamond in figures), then start a circular maneuver at V_F and ω_F and finally hold reached heading.

Conclusion and Future Works

In this paper we present a potential security threat in the conflict detection and resolution methods. Such vulnerabilities are easily enlightened in satellite-based network air traffic control. Using an evader-pursuer scenario, we showed that two pursuers might threat a collision with the evader, which will, consequently, change its original path to any desired dictated path. Using simulations and experiments, we show that particular knowledge of inner-working of collision and detection algorithm along with shared flight information allows malicious pursuers to systematically induce combinations of arcs and straight lines on autonomous crafts. This is considered a threat because control of a vehicle will be lost. This kind of unexpected effects and consequence should be further investigated in order to develop effective deconfliction methods for autonomous vehicles. In fact, a comprehensive study of the trustworthiness of an autonomous system should include study of potential abuses of the GNC algorithms used and their dependencies.

Most of the results presented here are obtained through a deterministic treatment of the problem. Analytical results defining margin of controllability on the evader represent a stimulating future development. This work can motivate enhancement in collision avoidance systems to make them more robust to unwanted trajectory alterations. For example, introducing a level of randomized behavior can help an aircraft recognize a correlation between its own trajectory and neighbors and could lead to identification of the threat.

References

[1] United States Government Accountability Office , April 2015, FAA Needs a More Comprehensive Approach to Address Cybersecurity As Agency Transitions to NextGen, AIR TRAFFIC CONTROL.

[2] Henn, Steve, 2012, August, npr.org. [Online], http://n.pr/1rDoP56

[3] King, Luis, 2003, Audit Announcement- Security Control of the Automatic Dependent Surveillance-Broadcast System, FAA.

[4] McCallie, Donald, Jonathan Butts and Robert Mills, 2011, Security analysis of the ADS-B implementation in the next generation air transportation system, International Journal of Critical Infrastructure Protection, vol. 4, no. 2, pp. 78.

[5] Thurber, Matt, September 2012, AINonline. [Online], http://bit.ly/1mK90mv.

[6] Pierpaoli, Pietro, Riccardo Zanforlin, Amir Rahmani, 2015, Vulnerability of UAV Sense and Avoid to Exploitations: Non-Cooperative Trajectory Modifications, AIAA Guidance, Navigation and Control Conference, Kissimmee, Florida.

[7] Kuchar, James, Lee C. Yang, 2000, A review of conflict detection and resolution modeling methods," Intelligent Transportation Systems, IEEE Transactions on, vol. 1(4), pp. 179-189.

[8] Albaker, Rahim, 2009, A survey of collision avoidance approaches for unmanned aerial vehicles, in Technical postgraduates (TECHPOS), international conference for, pp. 1-7.

[9] Fiorini, Paolo, Zvi Shiller, 1998, Motion planning in dynamic environments using velocity obstacles, The International Journal of Robotics Research, vol. 17(7), pp. 760-772.

[10] Van der Berg, Jun, Ming Lin, Dinesh Manocha, 2008, Reciprocal velocity obstacles for real-time multi-agent navigation, in Robotics and Automation ICRA, IEEE International Conference on, pp. 1928-1935.

[11] Kanayama , Yutaka, Yoshihiko Kimura, Fumio Miyazaki, Tetsuo Noguchi, 1990, A Stable Tracking Control Method for an Autonomous Mobile Robot, in Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH.

[12] Löfberg, Johan, 2004, YALMIP: A Toolbox for Modeling and Optimization in MATLAB, in In Proceedings of the CACSD Conference, Taipei, Taiwan.

[13] Shima, Tal, Steven Rasmussen, 2009, UAV cooperative decision and control: challenges and practical approaches, SIAM, Ed., vol. 18.

Acknowledgements

Authors want to thank Rowland O'Flaherty, Yancy Diaz and Maria Santos at GRITS lab, Georgia Institute of Technology who helped with experimental setup used in this paper.